Northumbria Research Link

Northumbria
University
NEWCASTLE

Evolutionary Multi-Agent Transfer Learning with Model-based Opponent Behavior Prediction

Yaqing Hou, Yew-Soon Ong, *Fellow, IEEE*, Jing Tang and Yifeng Zeng

***Abstract*—This paper embarks a study on multi-agent transfer learning for addressing the specific challenges that arise in complex multi-agent systems where agents have different or even competing objectives. Specifically, beyond the essential backbone of a state-of-the-art evolutionary Transfer Learning framework (eTL), this paper presents the novel Transfer Learning framework with Prediction (eTL-P) as an upgrade over existing eTL to endow agents with abilities to interact with their opponents effectively by building candidate models and accordingly predicting their behavioral strategies. To reduce the complexity of candidate models, eTL-P constructs a monotone submodular function, which facilitates to select Top-K models from all available candidate models based on their representativeness in terms of behavioral coverage as well as reward diversity. eTL-P also integrates social selection mechanisms for agents to identify their better performing partners, thus improving their learning performance and reducing the complexity of behavior prediction by reusing useful knowledge with respect to their partners' mind universes. Experiments based on a partner-opponent minefield navigation task (PO-MNT) have shown that eTL-P exhibits the superiority in achieving higher learning capability and efficiency of multiple agents when compared to the state-of-the-art multi-agent transfer learning approaches.**

***Keywords*—Multi-agent System, evolutionary transfer learning, behavior prediction, monotone submodular model selection.**

I. Introduction

Transfer learning (TL) has surfaced as an attractive learning approach for enhancing the learning efficacy of a new task by reusing the valuable data from a related task [1]. Due to its flexibility and ease of use, TL has been well-known as a popular problem solver that enjoyed its significant success across a wide realm of real-world applications including computer vision [2], natural language processing [3], etc. Recent study on TL has also started to investigate multi-agent systems (MASs) wherein agents tend to benefit from the knowledge transferred from their partners of high payoffs, hence improving agents' performance in more efficient problem-solving [4] [5] [6]. Emerging multi-agent TL approaches include Advice Exchange (AE) mechanisms [7], a Parallel Transfer Learning (PTL) approach [8] and an evolutionary Transfer Learning (eTL) framework [9].

Existing study on multi-agent TL approaches has to date focused on simple multi-agent scenarios where all agents have the same or share similar action spaces, and pursue a common objective. In multi-agent cases where different agents have competing objectives, the learning process of multiple agents becomes more complex and current TL approaches do not cope well, since agents are expected to be aware of the thoughts of their opponents so that efficient strategies can be developed to complete the mission successfully [10]. However, searching for an efficient interactive strategy is a challenging task since its effectiveness greatly depends on the behavior of the opponents involved. Therefore, it is necessary to endow agents with capacities to identify the strategies, capabilities or models of their opponents that are present in the competitive multi-agent environment [11].

The ability to accurately predict the actions of an intelligent agent is useful for a wide variety of problems, such as commercial video games [12] and automated driver assistance [13]. In competitive multi-agent scenarios, subject agents need to predict the action that the opponent agent will take in a given environmental state, or more specifically, to sample from the distribution over actions that is as close as possible to its actual action distribution. To achieve this, we can reasonably assume that opponent agent reasons about the actions it takes and hence tries to model that reasoning process as explicit as possible. Notably, the true model of opponent agents is usually unknown especially in a competitive multi-agent scenario [14]. Nevertheless, we do not need know exactly how the agent reasons about their actions, as many different processes may lead to the similar observed behavior. However, while different models might ultimately lead to the same optimal behavior, if those models are given certain computational bounds nor allowed to learn with sufficient data, they may yield potentially very different approximate solutions. Therefore, numerous models are often constructed to reason about the behavior distribution of opponent agents, based on previously recorded data on those agents' behavior [11]. The challenge here is hence to identify the one model which enables a more accurate prediction of opponent agents' actual behavior.

Taking this cue, the interest of the current work lies in the development of TL in more complex and realistic MASs where multiple agents have different or even competing objectives. Specifically, we present an enhanced multi-agent TL framework where subject agents could improve their learning performance by building numerous candidate models of their opponents and accordingly predicting their behaviors. In particular, considering that the search space of numerous models

Yaqing Hou is with the Data Science and Artificial Intelligence Research Centre (DSAIR), School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (e-mail: yaqinghou@ntu.edu.sg).

Yew-Soon Ong is with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore 639798 (e-mail: asysong@ntu.edu.sg).

Jing Tang is with the School of Computing, Teesside University, UK (e-mail: J.Tang@tees.ac.uk)

Yifeng Zeng is with the School of Computing, Teesside University, UK (e-mail: y.zeng@tees.ac.uk).

could be large [15], the identification of an appropriate model from the full model set becomes time-consuming and even infeasible due to the computational and memory limits. In the light of this consideration, we propose a Top-K model selection method to select a proper subset of candidate models from the full model space by measuring the representativeness of the selected models to the full model space. As the size of selected models is much smaller than the full model set, the search space and complexity of candidate models will be reduced significantly, hence improving the computational efficiency. The essential backbone of our proposed approach is the state-of-the-art eTL framework [9] which is driven by a series of memetic mechanisms [16] that are inspired from Universal Darwinism and Darkins' notion of meme(s) [17]. Further, we employ a well-established temporal difference (TD) - Fusion Architecture for Learning and Cognition (FALCON) [18] as the connectionist reinforcement learning agents in the present study. In particular, the core contributions of this paper can be summarized as follows:

1) A novel evolutionary multi-agent TL framework (eTL-P) is proposed for modeling subject agents in competitive multi-agent settings wherein subject agents are able to predict the behavior/actions of their opponents by solving the attributed candidate models.
2) eTL-P endows agents with the capacities to build candidate models automatically given the available data instances recorded from the historical interactive activities. In addition, eTL-P focuses on model complexity reduction and proposes a Top-K model selection method to select a proper subset of models by measuring their representativeness to the full model space. Taking the coverage and diversity aspects of selected models into consideration, the selected Top-K models are proven to be representative to the full model set, hence guarantee promising prediction accuracy.
3) eTL-P integrates social selection mechanisms for subject agents to identify their better performing partners in the environment while online. This leads to enhancement in learning performance and a reduction in the complexity of behavior prediction since subject agents benefit from the useful knowledge which they leverage from their partners' mind universes (i.e., both from the internal connectionist learners and predictive candidate models of opponent agents).
4) To validate the efficacy of the proposed eTL-P, we conduct comprehensive empirical studies on a Partner-Opponent Minefield Navigation Task (PO-MNT) involving both subject agent partners and their competing opponents. Empirical results show that eTL-P improves the learning effectiveness and efficiency of subject agents in the complex PO-MNT scenarios.

The rest of the paper is organized as follows. Section II presents an overview of multi-agent TL and agent modeling in competitive MASs which is followed by an introduction of the eTL framework. Section III discusses the comprehensive deta ils of the proposed eTL-P, which is composed of behavior prediction, Top-K model selection and multi-agent TL scheme. Subsequently, empirical study of the proposed eTL-P is investigated on an adapted MNT in Section IV. Last but not least, Section V presents the brief concluding remarks of this paper.

II. Preliminaries

This section begins with some background or overview of the multi-agent transfer reinforcement learning study, which is followed by an introduction of the eTL using (TD)-FALCON as the basic infrastructure of a reinforcement learning agent.

A. Background

1) Multi-agent Transfer Learning: Reinforcement learning (RL) is a paradigm for learning sequential decision making tasks that enable individual agents to learn and adapt to the environment [19] [20]. Typically, a decision making task is formulated as a Markov Decision Process (MDP) which is composed of a set of states S, a set of actions A, a reward function $R(s, a)$, and a transition function $P(s'|s, a)$. Given each state $s \in S$, the agent takes an action a from action set A. Upon receiving a reward $R(s, a)$ after performing this action, the agent arrives at a new state s' which is determined by the probability distribution $P(s'|s, a)$. A policy $\pi = P(a|s)$ specifies a distribution for each state on deciding which action an agent takes. The value $Q^*(s, a)$ of a state-action pair is an estimate of the future reward obtained from (s, a) when following policy π, and is determined by solving the Bellman equation:

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) max_{a'} Q^*(s', a')$$

where $0 < \gamma < 1$ is the discount factor. The goal of a RL agent is hence to find the policy π mapping states to appropriate actions that maximizes the expected long-term rewards obtained from environment.

RL has attracted extensive attentions in the past decades [21] [22] [20]. A plethora of RL methods, such as Monte Carlo [23], temporal difference [24] and direct policy search [25], have been proposed for building independent autonomous agents. Nevertheless, while existing RL methods have achieved significant success, they usually require a lot of experiential data and high exploration time to learn, hence are deemed to be slow and sometimes fall short in meeting with today's competitive need for high-efficiency problem-solvers in many complex problem domains.

On the other hand, TL has been gaining increasing attentions for enhancing the classical RL paradigm, which leverages the useful strategies from a well studied domain as supplementary knowledge to instruct learning process on newly encountered tasks [1]. Recently, a variety of TL methodologies, such as instance transfer [26], action value transfer [27] [28], model transfer [29] and advice exchanging [7] have been proposed, and benefited a wide range of RL tasks. However, it is worth noting that these approaches focus almost exclusively on speeding up learning across single agent systems [8]. The transfer among multiple agents while the learning progresses on-line in the same environment has less been considered.

One fundamental challenge which TL faces is that of gathering useful knowledge from necessary number of learning trials to learn correctly. In multi-agent settings, TL could be rather problematic because of the potentially much larger search space arising from the number of agents or their behavioral sophistication. Current research on TL is beginning to take steps towards addressing this specific challenge. Several studies have been investigated including the interactive AE mechanisms [7] in which agents with poor performance request behavioral advice from the elitist agent, a PTL approach [8] where agents broadcast valuable data instances to all other partners and an eTL framework [9] wherein the knowledge defined as meme(s) are transmitted to others via a human-like imitation process. Nevertheless, all of these research on multi-agent TL has focused on a simple multi-agent scenario where all agents share the same objectives and remained yet to fully exploit the common traits that exist in the different objectives.

When compared to state-of-the-art multi-agent TL learning approaches, eTL exhibits significant superiority to achieve greater level of adaptivity in addressing the increasing complexity of problem-solving. Previously, eTL has been successfully used for modeling autonomous agents in a commonly used MNT and a well-known first person shooter game "Unreal Tournament 2004" [9]. Besides, an interactive game, namely "Home Defence" has also been investigated where non-player characters driven by eTL interact naturally with human players [30]. Taking this cue, beyond the formalism of eTL, this paper contributes to the study of multi-agent TL and further embarks a novel study on the proposed eTL-P in addressing the challenges arising in complex MASs where agents have non-aligned, or even competing objectives.

2) Agent Modeling in Competitive Multi-agent Systems: For autonomous agents in MAS, the ability to reason about or predict the behavior of other agents is crucial to one's own performance [31]. Specifically, knowing the likely actions of other agents influences an agent's expected distribution over future environmental states, and thus informs its planning of future behaviors. In a competitive environment, the predicted behavior of other agents with differing objectives is referred to as an opponent model [32] [33]. Opponent models are particularly useful if they enable some identification of potential patterns or weakness on the part of the opponent. For example, a chess player can determine how best to play away from an opponent's strengths via studying past games of that opponent.

Generally, an opponent model is a function which takes as input some portion of the observed interaction history, and returns a prediction of the future actions regarding the opponent agent [11]. The interaction history may contain information such as the past actions that the opponent took in various circumstances. In the literature, an autonomous agent can construct such a model in different ways. Most of research work has endeavored to learn opponent models from scratch via policy reconstruction [12], which makes explicit predictions about an agent's actions by reconstructing the agent's decision making. These methods often begin with some arbitrary or idealized model and "fit" the internals of the model to reflect the agent's observed behavior [34]. Nevertheless, policy reconstruction can be a slow process, since numerous observations may be required before the modelling process yields a useful model. This tends to be problematic in scenarios in which an agent has neither time nor opportunity to collect sufficient observations about the opponent. In such cases, it is useful if an agent is able to reuse models learned in previous interactions with other agents, such that it only needs to find the model which most closely resembles the observed behavior of the opponent agent in the current interaction.

Based on the above motivation, type-based (or model-based) methods reason about a space of possible types that the opponent agent may have [35] [36]. Each type is a complete specification (or a model) of the agent's behavior, taking as input the observed interaction history and assigning probabilities to the actions available to the opponent agent. The representation of types can be formalized by decision trees and artificial neural networks, etc. Moreover, in historical studies, types may be obtained in different ways: they may be specified manually by a domain expert [14]; they may have been learned in previous interactions or generated from a corpus of historical data [37]; or they may be hypothesized from the domain and task to be completed [38]. Nevertheless, most of these studies on type-based opponent modelling methods assume that the specifications of types (e.g., artificial neural networks of specific hyper-parameter configurations) are available in advance. In case we have no prior knowledge on the specific representations of types, it is difficult or time-consuming to predict opponent behaviors with all possible candidate types.

The challenge here is hence to identify a small set of types (or models) from all candidates by solving which enables an accurate prediction of opponent agents' behavior. To this end, a Top-K model selection approach is proposed in this paper to reduce the search space or complexity of candidate types by selecting a proper subset of candidate types from the full model set.

B. eTL Framework

eTL takes inspiration from Universal Darwinism and Dawkins's definition of a meme. In meme inspired memetic computing, many of the existing work has been established as an extension of the classical evolutionary algorithms where a meme is perceived as a form of individual learning procedure or local search operator in population based search algorithms. Differing from these studies, the eTL framework, which is depicted in Fig. 1, introduces a more meme-centric learning framework which comprises a series of meme-inspired evolutionary knowledge representation and transfer mechanisms including *meme representation*, *meme expression*, *meme assimilation*, *meme internal evolution* and *meme external evolution*. In particular, meme(s) form the underlying building blocks (idea, knowledge, emotion, etc.) of the mind universe of an agent. In *meme representation*, meme(s) are knowledge stored internally in agents' mind universe (memotype) and transpired externally as the behavioral actions that can be transmitted to other agents via social interactions (sociotype). *Meme expression* performs a readout of the internal knowledge as observable actions while *meme assimilation* endows agent with capabilities to capture observable actions by other agents and updates them into the mind universe.

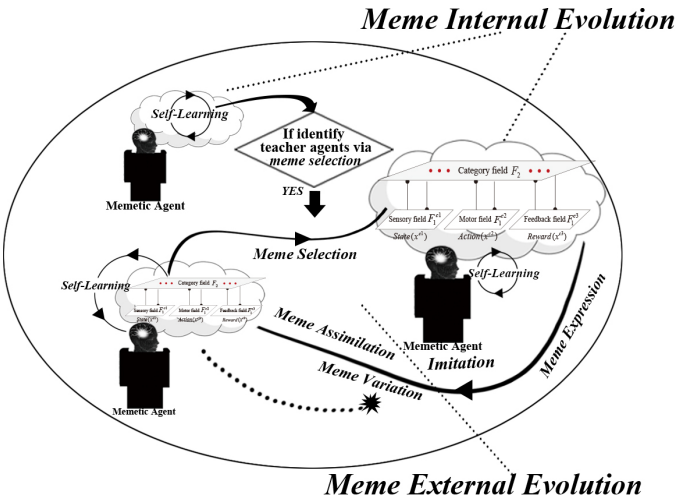


Fig. 1. Illustration of eTL wherein the mind universe of an agent takes the form of FALCON.

In our present study, we consider a manifestation of neuronal meme(s) as recurring patterns captured within neural networks that define the behaviors or guiding criteria of agents. Specifically, the mind universe of an agent takes the form of FALCON (depicted in Fig. 1). In particular, a FALCON employs a three-channel neural network architecture and comprises a category field F_2 for storing acquired memotypes and three input fields, namely, a sensory field F_1^{c1} for representing current states, a motor field F_1^{c2} for representing actions, and a field F_1^{c3} for representing reward values. All of the memotypes in F_2 form the knowledge of the agent which models the association from current states and action to the reward values. In particular, the baseline vigilance parameter $\rho^k \in [0, 1]$ *for* $k = 1, 2, 3$ is extremely important since it controls the level of match criterion on the state and action spaces so as to encourage knowledge generalization. Typically, increasing the vigilance values generally improves the predictive performance of FALCON agents with the cost of generating more category knowledge inside their mind universe.

Meme internal evolution and *meme external evolution* are central to the behavioral learning aspects of eTL. *Meme internal evolution* serves to update agents' mind universe by self learning. *Meme external evolution*, on the other hand, facilitates to model the social interaction among agents. In particular, the evolutionary knowledge transfer process is primarily driven by the imitation in a cultural evolution process. The overall framework of eTL is outlined in Algorithm 1. First, the crowd of agents in the environment is generated (line 1). Then, each agent undergoes *meme internal evolution* independently while operating in the environment (line 4). Meanwhile, *meme external evolution* proceeds whenever an agent identifies a teacher agent of high payoff via *meme selection* (line 6). Once the teacher agent is selected, *meme transmission* occurs to instruct how the agent imitates from others (line 9). During this process, *meme variation* facilitates the intrinsic innovation tendency of the transferred actions from the teacher agent. Notably, an agent makes use of gained knowledge from other agents via *meme assimilation* after *meme external evolution*. The current agent assimilates the transferred knowledge by undergoing the advised action from teacher agents under given environment states. In this way, the leaning machine of the agent could learn the association from current states and the imitated action to the reward values, thus providing instruction for future action prediction.

Algorithm 1: eTL Framework

```
1  Initialize: Generate all agents
2  while the stopping criteria is not satisfied do
3      for each current agent do
4          Perform meme internal evolution process
5          /*Meme external evolution*/
6          if identifies a teacher agent via meme selection
             then
7              Perform meme expression with teacher
                 agent given the state of current agent
8              Perform meme variation on the transmitted
                 knowledge with probability ν
9              Perform meme transmission to transfer
                 teacher's action to current agent
10             /*ν is the frequency probability of variation
                 process*/
11         /*Perform and learn the action from meme
             external evolution if identified teacher agent,
             else the action from meme internal evolution*/
12 End
```

III. Proposed eTL-P Framework

In this section, we present the proposed eTL-P which endows subject agents with capability to interact with their opponents effectively by predicting their strategies, in addition to the usual ability of competency of learning from friendly teachers agents. The proposed eTL-P framework (as shown in Fig. 2) is composed of model-based behavior prediction, Top-K model selection and multi-agent transfer learning, respectively. In particular, model-based behavior prediction approach proceeds to predict the behaviors of opponent agents based on the predictive candidate models trained using historical data instances, Top-K model selection aims to choose a proper subset of candidate models from the full model set and multi-agent TL enhances the learning capability of subject agents by leveraging beneficial knowledge transferred from their partner subject agents. In what follows, we first describe the objectives of multiple agents in a competitive multi-agent environment and then provide the detailed realization of each component in the proposed eTL-P.

A. Objectives of Multiple Agents in Competitive MAS

The reinforcement learning process of a single subject agent in eTL-P is formulated as a sequence of markov decision

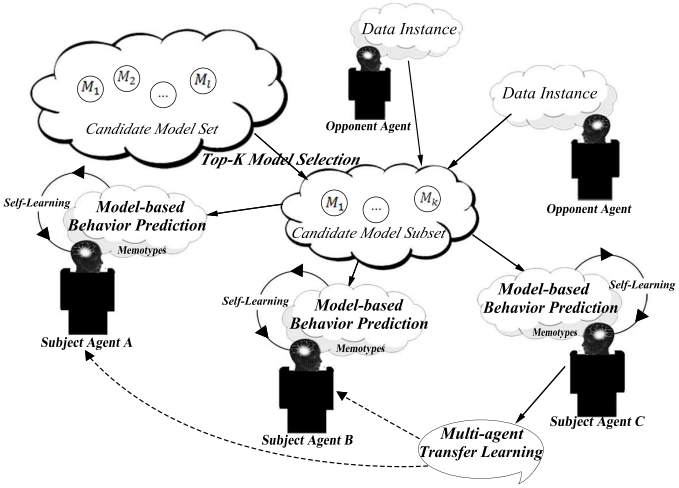


Fig. 2. An illustration of the proposed eTL-P.

processes (denoted as MDPs in Section II-A) which continue until the stopping criteria, such as mission numbers or fitness levels, are satisfied. All agents are equipped with a set of sensors so they have access to the environment states. Depending on the obtained states and the individual's knowledge, a subject agent learns to select and perform the most appropriate action, and updates its mind universe under the motivation of receiving positive rewards in the future. During the learning process, each subject agent encounters and interacts with other agents who may be potential partners (agents share a common objective), or hostile opponent agents (agents have competing objectives). All agents possess mind universes that are unknown to none other in the environment. In each time step, all subject and opponent agents operate in parallel within a common environment.

B. Model-based Behavior Prediction

In a competitive multi-agent environment, a subject agent requires an efficient strategy to counter opponent agents in order to complete the missions successfully. However, since a subject agent has no prior knowledge about the opponents, the design of suitable strategies to complete the missions can be non-trivial. The task of interests here is thus to endow subject agents with capabilities to predict the expected behaviors of the opponents.

Due to the scarcity of data availability in the environment under study, we consider the model-based (or type-based) [35] approach for predicting the behaviors of opponent agents. Specifically, in this work, the models (or types) are represented as neural network structures since both subject agents and their opponents employ neural networks as their RL machines. Further, since the true model of opponents (e.g., an artificial neural network of specific hype-parameter configuration) is not always approximated with high certainty, a number of potential candidate prediction models $M = \{m_1, m_2, ..., m_l\}$ of differing hype-parameters is constructed and thereafter assigned to subject agents.

Algorithm 2 summarizes the details of model-based behavior prediction process of subject agents. To begin, all of the candidate models M are expected to be trained using the historical data instances of opponent agents (lines 2-5). In our design, data instances of opponent agents are collected in advance and of a limited data size l. We employ instance-based model training method since it has been widely used for training RL models in the literature [39], [40]. Typically, data instances are formulated as a sequence of $< \mathrm{S}, \mathrm{A}, \mathrm{R} >$ tuples. Given each tuple, a candidate model learns to associate the state, action and reward and assimilates them into their internal learning structures. Once the candidate model training process completes, each subject agent will start to undergo self learning independently.

The behavior prediction of opponent agents occurs whenever a subject agent identifies an opponent agent $agt(o)$ nearby during the self learning process of subject agents (line 12). Once an opponent agent is detected, an ϵ-greedy model selection scheme proceeds to instruct how subject agent chooses an appropriate one from all attributed candidate models. This scheme aims to balance the fundamental trade-off between *exploitation*, i.e., sticking to the most believed model, and *exploration*, i.e., trying out other seemingly complementary models. Specifically, subject agent with ϵ-greedy scheme selects a candidate model m_s of the highest confidence level $conf(m_s)$ with probability 1 - ϵ ($0 \leq \epsilon \leq 1$), or otherwise chooses a random model with a probability of ϵ (lines 13-16). The value of ϵ decays gradually with behavior prediction time frames of subject agents.

Upon selecting a candidate model successfully, subject agent $agt(c)$ then predicts its virtual state $state(agt(c))'$ by attempting each available action a_i from the action set A (line 18). Furthermore, the state $state(agt(c))'$ is used to infer the corresponding virtual state $state(agt(o))'$ of opponent agent $agt(o)$. By activating the selected model m_s, subject agent $agt(c)$ is able to predict the action a_o of $agt(o)$ given $state(agt(o))'$ (line 19). Based on the results after virtual performing a_o by $agt(o)$, subject agent then decides whether to discard an action from the available action set A (lines 21 and 22). Subsequently, agent $agt(c)$ proceeds to select an action a_s from the updated A with the highest reward yield (line 23) and continues the self learning process. Notably, each candidate model is attributed with some level of confidence on its prediction accuracy. The model confidence level is updated along with the learning process, according to how well candidate models performed on predicting the behaviors of opponent agents. This ensures that subject agents can adapt accordingly whenever opponent agents change their strategies on-line. The m_s leading to the successful behavior prediction is updated with a higher confidence $conf(m_s)$ of being chosen for the subsequent behavior prediction process. On the contrary, when leading to the failure behavior prediction, the model will be updated with a lower confidence level. (line 25):

$$\begin{cases} conf(m_s) = conf(m_s) + Reward, & \text{if succeeds} \\ conf(m_s) = conf(m_s) - Penalty, & \text{if fails} \end{cases}$$

where reward and penalty are defined as positive integers and $conf(m_s) \in [-100, 100]$.

Algorithm 2: Learning with Behavior Prediction

Input: Candidate models $M = \{m_1, m_2, ..., m_l\}$, Collected data $D = \{D_1, D_2, ...D_v\}$, where $D_v =< S_v, A_v, R_v >$, Action set A
Output: Subject agents $\{agt(c)\}$

Begin:
for *all* $M_l \in M$ **do**
  for *all* $D_v \in D$ **do**
    Set input vector $< S_v, A_v, R_v >$ in M_l
    Training with vector $< S_v, A_v, R_v >$
Generate initial subject agents $\{agt(c)\}$ and their opponents $\{agt(o)\}$
Assign each subject agent with candidate models M
while *the stopping criteria are not satisfied* **do**
  for *each current subject agent* $agt(c)$ **do**
    Perform *meme internal evolution* process
    /*Behavior Prediction*/
    if *detect an opponent agent* $agt(o)$ **then**
      if $Rand > scheme(\epsilon)$ **then**
        $m_s = Random\{m_s : for\ all\ model\ M\}$
      else
        $m_s = Max\{conf(m_s) : for\ all\ model M\}$
      for *each available action* $a_i \in A$ **do**
        $state(agt(c))' =$ *Virtual_Perform*$(agt(c), a_i)$
        $a_o = Predict(m_s, state(agt(c))')$
        $state(agt(o))' =$ *Virtual_Perform*$(agt(o), a_o)$
        if *fail interaction after virtual move* **then**
          Discard a_i from action set A
    Get a_s from A with highest reward of self learning in $agt(c)$
    Continue *meme internal evolution* process
  Update $conf(m_s)$
End

C. Top-K Model Selection

Ideally, subject agents with candidate models derived from RL data instances could approximately predict the true behavior of opponent agents if the true model of their opponents is available in the candidate model space. However, to achieve such result becomes computational intractable due to the high complexity of candidate models ascribed to opponent agents and thus ways of mitigating the computational intractability are critically expected. Since the complexity is predominantly due to the space of or cost on solving candidate models, we therefore endeavor to provide reasonably effective polices by selecting a proper subset of candidate models given the limited model space while avoiding a significant loss in optimality.

In the past few years, researchers have concentrated on compressing the candidate model space of other agents. Among them, some works focus on selecting a subset of models that are expected to have the largest joint coverage of solutions of all candidate models [41]. By taking this coverage into consideration, the behavioral information loss is reduced effectively. The second category, on the other hand, aims to select models that are concise and contain as few redundant models as possible [42]. Accordingly, models that provide similar information of opponent agents are not expected to be selected concurrently. In practice, the redundancy can be effectively reduced by enforcing the diversity among the selected models. Taking these inspirations, we initiate a novel Top-K model selection scheme which considers a fusion of coverage and diversity principles to select the representative candidate models.

Formally, our task of model selection is to select a subset $M^K \subseteq M$ for representing the full model set M. Obviously, the size of M^K shall be much smaller than M and is expected to be restricted within the given model size budget K. Such constraints on M^K can be modeled as *knapsack constraints*: $\sum_{m_i \subseteq M^K} c_{m_i} \leq K$ where c is the non-negative cost of collecting m_i. If we employ a set function $\mathcal{F} : 2^M \rightarrow \mathbb{R}$ for measuring the quality of M^K, we can summarize the model selection task as a combinatorial optimization problem in Eq. (1).

$$M^{K*} \in \underset{M^K \subseteq M}{\text{argmax}} \mathcal{F}(M^K) \text{ subject to : } \sum_{m_i \in M^K} c_{m_i} \leq K. \quad (1)$$

As this is a generalization of the cardinality constraint where $\forall m_i, c_{m_i} = 1$, the combinatorial optimization problem constitutes a well known NP-hard maximum coverage problem [43]. Particularly, taking both of the abovementioned coverage and diversity into consideration, the specific definition of our $\mathcal{F}(M^K)$ is further formulated as the following.

$$\begin{aligned} &\textbf{Given} : M, K \\ &\textbf{Objective} : \\ &max_{M^K \subseteq M, |M^K| = K}, \mathcal{F}(M^K) = \sum_{i=1}^{K} \sqrt{\sigma(P_i \cap M^K)} \end{aligned} \quad (2)$$

where P_i is a partition of the full model set M into separate clusters and $\sigma(P_i \cap M^K)$ is the estimated coverage probability of the selected models $P_i \cap M^K$ to the full model space M. Applying the square root $\sigma(P_i \cap M^K)$ further rewards diversity in that there is higher payoff of selecting a model from a cluster not yet having one of its units already chosen. Notably, the comprehensive discussion on coverage and diversity objectives shall be found in what follows.

In the literature, the optimization of maximum coverage problem is deemed to be approximately solved when $\mathcal{F}$ is monotone submodular using generic greedy forward selection algorithms [44]. Especially, submodularity and monotonicity are two necessary ingredients to guarantee that such greedy algorithms give near-optimal solutions.

Here, we provide the detail of $\mathcal{F}(M^K)$ and prove it to be monotone and submodular.

1) Coverage: Firstly, suppose $M' = P_i \cap M^K$, $\sigma(M')$ is integrated for representing the coverage of M' by M. Existing study on the possible $\sigma(M')$ has reported several ways for representing the coverage of M'. For instance, $\sigma(M')$ could be facility location function [45], $\sigma(M') = \sum_{m_j \in M} max_{m_i \in M'} \sigma(m_i, m_j)$ or the graph cut function [44], $\sigma(M') = \sum_{m_j \in M \backslash M'} \sum_{m_i \in M'} \sigma(m_i, m_j)$, where $\sigma(m_i, m_j)$ represents the similarity between m_i and m_j. In our approach, $\sigma(M')$ is defined in a rather simple and common-used manner as the following:

$$\sigma(M') = \sum_{m_j \in M} \sigma(M', m_j) \tag{3}$$

where $\sigma(M', m_j)$ is computed as the proportional degree that m_j is covered by the model set M' in Eq. (4).

$$\sigma(M', m_j) = 1 - \prod_{m_i \in M'} [1 - \sigma(m_i, m_j)] \tag{4}$$

where $\sigma(m_i, m_j)$ indicates the degree to which m_j is represented by m_i from M', or in detail is calculated by how similar the behavior predicted by m_i is to that of m_j in Eq. (5).

$$\sigma(m_i, m_j) = \frac{1}{N} \sum_{t_{m_i} \in T_{m_i}, t_{m_j} \in T_{m_j}} \lambda(t_{m_i}, t_{m_j}) \tag{5}$$

where t_{m_i} is the state-action tuple from a set of predicted instances T_{m_i}, N is the size of T_{m_i} and $\lambda(t_{m_i}, t_{m_j})$ counts the number of identical action predictions given same states for T_{m_i} and T_{m_j}. In what follows, we prove the monotone submodularity of $\sigma(M')$.

Proof. 1. The coverage function $\sigma(M')$ *is monotone and submodular.*

Let S be a finite set. A submodular function $\mathcal{F} : S \to \mathbb{R}$ typically satisfies the property of *diminishing returns*: for any $A \subseteq B \subseteq S$ and $s \in S \backslash A$, $\mathcal{F}(A \cup s) - \mathcal{F}(A) \geq \mathcal{F}(B \cup s) - \mathcal{F}(B)$. In this case, the incremental value of s decreases as the context in which s is considered to grow from a smaller set A to a larger set B. In addition, a submodular function $\mathcal{F}$ is called monotone nondecreasing if $\forall A \subseteq B$, $\mathcal{F}(A) \leq \mathcal{F}(B)$.

In what follows, we first prove the monotonicity of function $\sigma(M')$. Let $M'_1 \subseteq M'_2 \subseteq M'$ and $m'_i \in M' \backslash M'_1$, $\sigma(M')$ satisfies:

$$\begin{aligned}
&\sigma(M'_1 \cup m'_i) - \sigma(M'_1) \\
&= \sum_{m_j \in M'} \sigma(M'_1 \cup m'_i, m_j) - \sum_{m_j \in M'} \sigma(M'_1, m_j) \\
&= \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)] - \prod_{m_i \in M'_1 \cup m'_i} [1 - \sigma(m_i, m_j)] \\
&= \sigma(m'_i, m_j) \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)]
\end{aligned} \tag{6}$$

As both $\sigma(m'_i, m_j)$ and $\sigma(m_i, m_j)$ are within the range of [0,1], we have $\sigma(M'_1 \cup m'_i) - \sigma(M'_1) \geq 0$. Therefore, $\sigma(M')$ is monotone. Further, the proof on the submodularity of $\sigma(M')$ is given by Eq. (7).

$$\begin{aligned}
&\sigma(M'_1 \cup m'_i) - \sigma(M'_1) - (\sigma(M'_2 \cup m'_i) - \sigma(M'_2)) \\
&= \sigma(m'_i, m_j)(\prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)] - \prod_{m_i \in M'_2} [1 - \sigma(m_i, m_j)]) \\
&= \sigma(m'_i, m_j) \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)](1 - \prod_{m'_i \in M'_2 \backslash M'_1} [1 - \sigma(m_i, m_j)])
\end{aligned} \tag{7}$$

Since $\sigma(m'_i, m_j)$, $\sigma(m_i, m_j)$, $\sigma(m_i, m_j) \in [0, 1]$,

$$\sigma(m'_i, m_j) \prod_{m_i \in M'_1} [1 - \sigma(m_i, m_j)](1 - \prod_{m'_i \in M'_2 \backslash M'_1} [1 - \sigma(m_i, m_j)]) \geq 0. \tag{8}$$

Therefore, $\sigma(M'_1 \cup m'_i) - \sigma(M'_1) \geq \sigma(M'_2 \cup m'_i) - \sigma(M'_2)$. $\sigma(M')$ is submodular.

Intuitively, $\sigma(M')$ is monotone since the model coverage always increases with a larger model set. On the other hand, $\sigma(M')$ is submodular: with two model sets A and B where $|A| < |B|$, the increment when adding a new model to A shall be larger since the information exhibited by the new model might have already been covered by those models that are in the larger model set B yet not in A. This is known as the exact property of *diminishing returns*.

2) Diversity: As separate model clusters P_i could represent distinct behaviors expected from candidate models, $\mathcal{F}(M^K)$ rewards diversity of M^K by encouraging to select models from different disjoint P_i of the full model set M ($\cup_i P_i = M$). In order to generate P, we employ a k-medoids algorithm to cluster M. Particularly, k-medoids is a clustering algorithm similar to k-means, but chooses a candidate model $m_j \in P_i$ with highest similarity value $\sum_{m'_j \in P_i} \sigma(m_j, m'_j)$ as the center of each cluster P_i. As soon as an element is selected from a cluster, other elements from the same cluster start having diminishing gain because of the square root function. For example, suppose we have $m_1, m_2 \in P_1$, $m_3 \in P_2$ and $\sigma(m_1) = 3$, $\sigma(m_1 \cup m_2) - \sigma(m_1) = 2.5$ and $\sigma(m_1 \cup m_3) - \sigma(m_1) = 2$. If m_1 is already in M^K, greedily selecting the next shall be m_3 rather than m_2 since $\sqrt{3 + 2.5} < \sqrt{3} + \sqrt{2}$.

Proof. 2. The function $\mathcal{F}(M^K)$ *is monotone and submodular.*

As the submodular functions possess properties in common with concave and convex functions, including the applicability and generality under a series of common operators such as mixtures, truncation, complementation, or certain convolutions [46]. We therefore conclude that if f is non-decreasing concave and $\mathcal{F}$ is nondecreasing submodular, the composition function $\mathcal{F}'(A) = f(\mathcal{F}(A))$ is nondecreasing submodular. In our case, the square root is a monotone concave function. Inside each square root exists a monotone submodular function $\sigma(M')$. Therefore, applying the summing square root to non-negative $\sigma(M')$ again yields a submodular function.

Summing up, the model selection function $\mathcal{F}(M^K)$ is both monotone and submodular. In particular, $\mathcal{F}(M^K)$ considers a fusion of coverage and diversity principles; hence it facilitates the selection of models that are fine representatives of each cluster while also being positively diversified across the multiple clusters. In addition, since $\mathcal{F}(M^K)$ is a complex optimization problem, we can also prove its NP-hard characteristic by simply converting $\sigma(M')$ in $\mathcal{F}(M^K)$ into the *budgeted maximum coverage problem (BMCP)* [43] by setting a budget of $L = K$, a collection of sets $S = M$ and associated unit cost of $c = 1$. The goal of BMCP hence is to find a collection $S' \subseteq S$ such that the total weight covered by S' is maximized given a budget L and a collection of sets S defined over a domain of weighted elements X. Since BMCP is NP-hard, $\mathcal{F}(M^K)$ is NP-hard as well.

To recap, if $\mathcal{F}$ is monotone submodular and NP-hard, it has been established that greedy algorithms can solve the combinatorial optimization problem in Eq. (1) at near-optimally with a $(1 - 1/e)$-approximation of the optimal solution. Therefore, we further propose Algorithm 3, which occurs prior to Algorithm 2 and employs a direct greedy method to optimize the model selection process. Particularly, the algorithm starts with $M^K = \emptyset$ and computes the $\mathcal{F}(M^K)$ for the selected models (line 2-5). Then, it iteratively adds the model that yields the greatest increment of $\mathcal{F}(M^K)$ until the model size of $|M^K| = K$ (line 6) is reached. After Top-K model selection, a proper subset of models M^K is selected out with the high behavioral representativeness of the full model space M. The complexity for learning the candidate models hence reduces from $|M|$ to $|M^K|$ ($|M^K| \ll |M|$).

Algorithm 3: Top-K Model Selection

1 **function**: $ModelSelection(M, K)$
2 $M^K = \emptyset$
3 **for** *Ite =1 to* K **do**
4 **for** ***all*** $m_i \in M$ **do**
5 $m_i \leftarrow argmax_{m_i}[\mathcal{F}(M^K \bigcup m_i) - \mathcal{F}(M^K)]$
6 $M^K \leftarrow M^K \bigcup m_i$
7 **return** M^K

D. Multi-agent Transfer Learning

To date, most of the existing works have focused on reducing the candidate model complexity for decision making problems, where a single agent predicts the behavior of another single one, but have yet to study the behavior prediction in a multi-agent setting [41]. Differing to existing approaches, we further propose a multi-agent TL approach for reducing subject agents' candidate model complexity with respect to the model space while also enhancing their learning capability.

The TL process between agents with unique learning capabilities is mainly driven by a human-like imitation process. In our multi-agent learning problems, all of the subject agents learn in the same environment and share a common behavioral action space. Therefore, the imitation-driven knowledge transfer process offers the advantage where imitating agents behave at approximately the same level of performance as their target of imitation. Further, when multiple candidate models are available, subject agents with the proposed TL approach are not required to train all the models and they can acquire increasing level of predictive capability from their partners by sharing predicting information with respect to their unique candidate models. TL approach thus reduces candidate model complexity of subject agents by attributing them with a comparatively smaller subset of candidate models.

Algorithm 4 summarizes the multi-agent TL approach in eTL-P. To begin, each subject agent is attributed with a set of candidate models $M^{agt} = \{M_1^{agt}, ..., M_q^{agt}\}$, where agt indicates the index of subject agent and q is the number of models in M^{agt} (line 2). Further, the current agent $agt(c)$ is expected to check whether there exists a better-performing teacher agent $agt(s)$ that has a higher fitness value $Fit(agt(s))$ (line 4). Once a teacher agent is identified, the current agent $agt(c)$ first passes its state $state(agt(c))$ to the teacher agent $agt(s)$ (line 6). Meanwhile, if agent $agt(c)$ detects an opponent agent $agt(o)$ nearby, it further transfers the inferred state $state(agt(o))$ of opponent agent to the teacher $agt(s)$ (lines 7-8). By simulating the states $state(agt(c))$ and $state(agt(o))$, the teacher agent $agt(s)$ undergoes behavior prediction of agent $agt(o)$ based on its own attributed model M^s and further expresses a predicted action a_s, which is observable and can be transmitted to agent $agt(c)$ (referred to Algorithm. 2). Then, if the transmitted action a_s is available, agent $agt(c)$ is expected to imitate and assimilate the action a_s into its own mind universe (line 13). In addition, a variation process with a probability ν is defined to keep the innovative diversity of selected actions therefore preventing agent $agt(c)$ from learning from teacher agents blindly during the multi-agent TL process (line 12).

Given the selected candidate model set M^K after Top-K model selection, multi-agent TL approach further partitions M^K into different partial sets and each set is labeled with the unique agent index: $M^K = \{m_1^1, m_2^1, ..., m_q^1, ..., m_K^{agt}\}$ where M^{agt} aggregates the candidate model set with the same subject agent superscript. In this way, subject agents merely need to train candidate models from their corresponding partial sets. According to the aforementioned TL approach, the non-familial peer subject agents with unique candidate models M^{agt} construct social interaction with one another so as to benefit from the knowledge transferred from the others (see Algorithm 4). In this manner, the proposed eTL-P offers significantly reduction in the candidate model space of each subject agent in the competitive multi-agent setting.

IV. Empirical Study and Analyses

In the literature, Minefield Navigation Task (MNT) has been regarded as a popular platform to testify the capacities of autonomous agents [9] [18]. Taking this cue, we investigate the effectiveness and efficiency of the proposed eTL-P under more complex scenarios of the MNT (as depicted in Fig. 3) where multiple agents have shared or competing objectives.

Algorithm 4: Multi-Agent Transfer Learning

Input: Candidate models $M = \{M^1, M^2, ...M^{agt}\}$, where $M^{agt} = \{m_1^{agt}, ..., m_q^{agt}\}$

Begin:
Allocate each subject agent with candidate model M^{agt}
for *each subject agent* $agt(c)$ **do**
  Perform Algorithm 2, line 10-22
  if *identify* $\{agt(s)|Fit(agt(c) < Fit(agt(s))\}$ **then**
    Get $state(agt(c))$ of current $agt(c)$
    Pass $state(agt(c))$ to teacher agent $agt(s)$
    if *detect an opponent agent* $agt(o)$ *nearby* **then**
      Pass $state(agt(o))$ to agent $agt(s)$
      Perform *Behavior Prediction* with M^s
      Discard failure actions from action set *A*
    Get a_s from A with highest reward of self learning in $agt(s)$
    Perform variation on a_s with probability ν
  Get a_s from $agt(s)$, otherwise get a_s from A with highest reward of self learning by $agt(c)$
  Continue left steps that are the same as learning process in Algorithm 2
End

A. Experimental Platform and Configuration

In the classical MNT, subject agents (denoted by green tanks in Fig. 3, namely *navigators* hereafter) share a common objective, which is to navigate across the minefield so as to arrive at the target position (denoted by a red flag and is randomly generated per mission) within a specified time frame, while avoiding collision with any mines nor other agents. In contrast to the classical MNT where only subject agents with a common objective are involved, here our interest is on minefield navigation problems (namely PO-MNT in short) comprising both navigator partners and competing opponents (denoted by red tanks, namely referred to *predator* hereafter) in the environment. Particularly, the predators are introduced with the objective of capturing navigators before navigators flee to the target place. Further, in our design, predators are equipped with a thick armor that always cannot be destroyed by mines.

All mobile agents (including navigators and predators) share a common action space, including turning left or right, moving forward and proceeding diagonally left or right. They are equipped with sonar sensors and hence have access to a set of observations about the minefield environment, including mine detection, navigator detection, predator detection and target bearing detection. Further, a navigator is rewarded with a positive reward of '1' if it arrives at the target successfully. Otherwise, it is assigned with a '0' reward. In the experiment, we consider a crowd of 6 autonomous agents with 6 mines and 1 target position that are randomly generated across

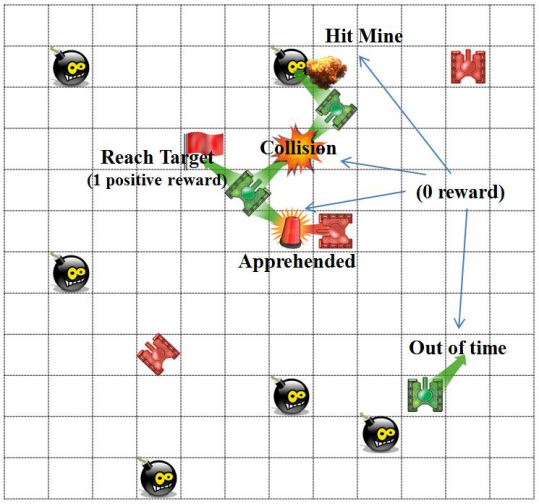


Fig. 3. Illustration of the adapted MNT (PO-MNT).

missions within a 12x12 minefield navigation environment. For each experiment, 30 independent simulation runs have been conducted and with each run involving a total of 5,000 randomly generated missions. One mission completes only if all navigators arrive at the target, hit the mines, exceed the maximum time steps of 30 step times or when they are apprehended by the predators.

Notably, predators are trained beforehand with the knowledge of apprehending their nearby navigators. On the other hand, navigators begin with zero knowledge but learn along with the missions undertaken. When navigators detect the nearby opponents via sonar sensors during the learning process, they can predict the behavior of predators using the proposed model-based behavior prediction method. Since both navigators and predators employ FALCON with $\rho^k = (0.2, 0.2, 0.5)$ as the connectionist reinforcement learning models that form their mind universe, the predictive predator models are formulated as a set of FALCON dynamics with different baseline vigilance values $\rho^k \in [0, 1]$ and further generated with data instances of predators that are collected from the online activities of the missions encountered. This configuration is attributed to the great importance of baseline vigilance parameter as it controls the knowledge generalization level of FALCON dynamics which affects their performance significantly.

The parameter configurations of (TD)-FALCON and the proposed eTL-P used in the present experimental study are summarized in Table I. For the purpose of a fair comparison, the configurations are maintained to be consistent with previous studies in [9] [47] [48]. To study the efficacy of the proposed eTL-P in solving the complex PO-MNT, comprehensive empirical results with respect to the following metrics are investigated:

- SR: the average success rate of agents on completing the missions;
- MT: the training time of candidate models that are

TABLE I. PARAMETER SETTING IN THE PROPOSED ETL-P.

Falcon Parameters	
Choice Parameters $(\alpha^{c1}, \alpha^{c2}, \alpha^{c3})$	$(0.1, 0.1, 0.1)$
Learning Rates $(\beta^{c1}, \beta^{c2}, \beta^{c3})$	$(1.0, 1.0, 1.0)$
Contribution Parameters $(\gamma^{c1}, \gamma^{c2}, \gamma^{c3})$	$(0.5, 0.5, 0)$
Baseline Vigilance Parameters $(\rho^{c1}, \rho^{c2}, \rho^{c3})$	$(0.2, 0.2, 0.5)$
Temporal Difference Learning Paramters	
TD learning rate α	0.5
Discount Factor γ	0.1
Initial Q-value	0.5
ϵ-greedy Model Selection Parameters	
Initial ϵ value	1
ϵ decay rate	0.0005
Demonstration Variation Parameter	
Frequency of Variation ν	0.1

assigned to navigators;

- PN: the number of prediction during the learning process;
- PT: the computational time cost for behavior prediction during the learning process.

B. Effectiveness of eTL

We begin with an investigation on the performance of eTL with 6 FALCON navigators on completing the classical MNT. This serves as a baseline for comparison. Then, two current state-of-the-art multi-agent TL approaches, namely, the Advice Exchange (AE) mechanisms [7] and Parallel Transfer Learning (PTL) [8] are considered in the experiment for the purpose of comparison. Specifically, AE mechanisms facilitate poor-performing agents to learn from their elite partners by seeking its advice on given circumstances. On the other hand, agents in PTL learn from others by leveraging the knowledge broadcasted by all others.

The resultant success rates (SRs) of the navigators for the different above-mentioned TL approaches on completing the classical MNT are denoted in Fig. 4. In particular, the average SRs of all 6 FALCON navigators on completing the classical MNT across the increasing number of missions are depicted in the figure. It can be observed that the SRs of the navigators for all TL approaches are noted to increase steadily right from the beginning of the navigation mission. Both eTL and AE outperformed the conventional MAS (Conv-MAS) as expected, since the latter has no mechanism for knowledge transfer. Further, eTL obtained superior SR over the other counterparts considered as the navigators are noted to learn better with increasing missions as compared to the existing state-of-the-art multi-agent TL approaches.

C. Results of eTL-P with True Predator Model

In what follows, we embark an investigation on the proposed eTL-P in addressing more complex PO-MNT wherein navigators and predators have competing objectives. In our study, a mixture of 3 navigators and 3 predators is considered in the PO-MNT. In the present set of experiments, we assume all navigators have access to the true model of the predators. In this manner, the navigator agents are able to predict the actions

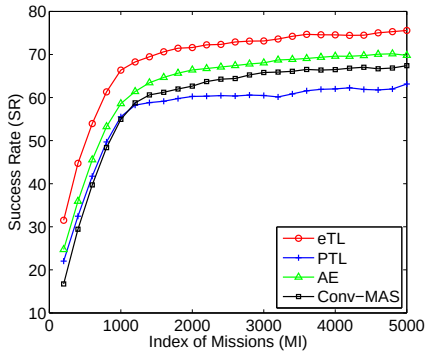


Fig. 4. SRs of navigators under eTL, PTL, AE and Conv-MAS on completing the missions in classical MNTs.

of the predators accurately during their navigation. Such an assumption and experimental study is carried out here to assess and validate the motivation for endowing agents with capacities to infer predators' actions as a means to improve their problem solving capabilities under situations where agents possess differing goals or competing objectives.

To begin, the performance efficacy of eTL-P is benchmarked against the conventional MAS (Conv-MAS) in which navigators have neither behavior prediction nor any transfer learning capability. Further, to study the effectiveness of each mechanism available in eTL-P, the performances of the agent navigators with only model-based behavior prediction capability (Mod-P), i.e., no TL capability, and agent navigators with only TL capability (eTL, referred to [9]), i.e., no behavior prediction capability, are also investigated, respectively.

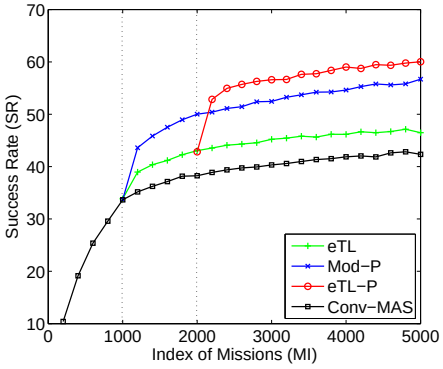


Fig. 5. SRs of eTL, Mod-P, eTL-P and Conv-MAS on completing the missions in PO-MNTs.

Fig. 5 summarizes the performance of eTL, Mod-P, eTL-P and Conv-MAS in terms of their respective SRs averaged at 100-mission intervals, across a total of 5,000 missions. Particularly, to identify the effect and significance of each mechanism in eTL-P, we first study the performance of eTL (which assumes navigators with only TL capability) and Mod-P (which assumes navigator with only opponent behavior

prediction capability) against Conv-MAS (i.e., no form of TL nor behavior prediction of opponents) from 1,000 missions. Subsequently eTL-P is pitted against eTL from mission 2,000 onwards, where both the capabilities of transfer learning and true model of opponent behavior are involved.

The SR results obtained showed that the performance of the navigators in Conv-MAS deteriorates significantly in the PO-MNT, i.e., Minefield Navigation Task with friendly partners and offensive opponents as compared to that in the classical MNT where there are no opponents in the latter task (see Section IV-B). This is understandable since navigators in PO-MNT now suffer from bad encounters with predators, i.e., they can be apprehended by their opponents, as compared to the classical MNT. Mod-P, which has the behavior prediction capability of opponents on the other hand, exhibits significantly higher SR over Conv-MAS. This indicates that, with the availability of behavior model of predators, navigators are shown to survive better in the PO-MNT. Notably, the results showed that navigators in the proposed eTL-P are observed to achieve superior performance in terms of SR at the end of the learning process, i.e., attaining a superior SR of around 63% at the end of 5,000 missions. This result highlights the efficacy of the proposed eTL-P in improving the learning performance of navigators on completing the PO-MNT successfully. Moreover, navigators in both eTL and eTL-P reported superiority in terms of SRs than ones without the multi-agent TL scheme. This can clearly be attributed to the transfer learning scheme of eTL and eTL-P which endows navigators with capacities to benefit from the knowledge transferred from their better performing partners, thus accelerating the learning speed and improving the SR of completing the missions.

D. Results of eTL-P with Predictive Predator Models of FALCON Dynamics

In previous subsections, a study on eTL-P wherein each navigator is assumed to have access to the true model of the opponent behavior is considered. In this subsection, we consider a more realistic scenario where navigators only have access to predictive candidate models of opponents as opposed to the true model of predators which is typically unknown in practice.

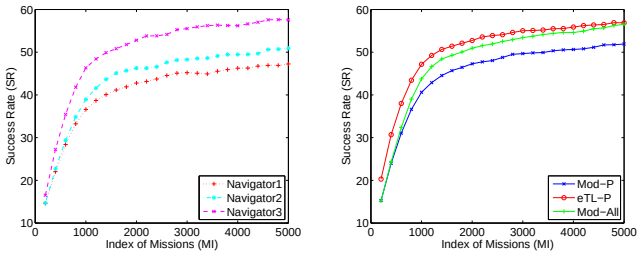
Given a set of predictive models, we assign each navigator with a differing subset of all candidates. The performance of navigators with attributed candidates is then reported by Mod-P in which navigators only have opponent behavior prediction capability, i.e., no multi-agent TL scheme. Since navigators with unique predictive models are likely to perform differently, the efficacy of eTL-P is therefore validated in enhancing Mod-P by facilitating the knowledge transfer across multiple navigators. In addition, we have also constructed experiments where each navigator is assigned with all the possible predictive models (denoted by Mod-All). The aim in designing such experiments were to assess the efficacy of opponent behavior prediction in choosing an appropriate model from multiple predictive candidates during the navigators' learning process. Based on these experimental settings, we study the performance of all mechanisms in eTL-P including model-based behavior prediction, multi-agent TL as well as Top-K model selection in what follows.

1) Performance of model-based behavior prediction and multi-agent TL: In this set of experiments, we employ three FALCON models as the predictive candidates, where each is configured with a differing vigilance level ρ^k, known as *1)* a low parameter $\rho^k = [0.1, 0.1, 0.1]$, *2)* a medium parameter $\rho^k = [0.4, 0.4, 0.4]$ and *3)* a high parameter $\rho^k = [0.9, 0.9, 0.9]$, respectively. Hence, each navigator in Mod-P and eTL-P is attributed with a unique FALCON candidate for opponent behavior prediction. The complete results pertaining to the SR of the navigators on the PO-MNT are summarized in Table II and Fig. 6. The analyses of the obtained results shall be discussed comprehensively next.

TABLE II. PERFORMANCE COMPARISON AMONG MOD-P, ETL-P AND MOD-ALL IN TERMS OF SR.

	FALCON	SR		
#	(ρ^1, ρ^2, ρ^3)	Mod-P	eTL-P	Mod-All
1	(0.1,0.1,0.1)	47.25	55.85	56.54
2	(0.4,0.4,0.4)	50.93	56.15	
3	(0.9,0.9,0.9)	57.56	58.63	

Fig. 6(a) depicts the performance of each navigator in Mod-P in terms of success rate. As can be observed, navigators with FALCON candidates of differing ρ^k parameters exhibited distinct SRs throughout the learning process. Among all three candidate models, Mod-P with FALCON of $\rho^k = (0.9, 0.9, 0.9)$ obtained the highest SR (57.56%, see Table II, column 3), significantly higher than the other two navigators (known as 47.25% and 50.93%, see Table II, column 3). This difference among navigators hence highlights the significance of choosing a proper candidate model for navigators to predict the behavior of their opponents. On the other hand, Mod-All (where each navigator is attributed with all predictive FALCON candidates) reported a competitive SR of 56.54% (see Table II, column 5) against the best performing Mod-P with a high vigilance value (57.76%, see Table II, row 3, column 3). This notable performance of Mod-All hence validates the efficacy of the proposed model-based behavior prediction method in selecting the appropriate model from all attributed candidates for opponent behavior prediction.



(a) Navigator1,2&3 in Mod-P. (b) Mod-P, eTL-P & Mod-All.

Fig. 6. SRs of Mod-P, eTL-P and Mod-All on completing the missions in PO-MNTs.

Further, Fig. 6(b) depicts SRs of Mod-P, eTL-P and Mod-All. As a result, both eTL-P and Mod-All obtained higher SRs when compared to Mod-P. In particular, without being attributed with all predictive FALCON candidates, navigators with the knowledge transfer scheme of eTL-P is noted to obtain SR that is slightly higher than Mod-All. It is worth noting that, under the eTL-P framework, navigators with FALCON candidate of $\rho^k = (0.1, 0.1, 0.1)$ achieved a significant SR increase of 8.6% as compared to that under Mod-P at the end of the missions (see Table II, row 1). As we discussed, this can be attributed to the navigators with multi-agent TL scheme benefiting from the knowledge by their better performing partners, such as navigators with FALCON candidate of $\rho^k = (0.9, 0.9, 0.9)$ in this case.

Moreover, in order to evaluate the performance of eTL-P on reducing the complexity of candidate models, the computational cost taken by eTL-P and Mod-All are summarized in Table III. Particularly, MT is referred to the training time of candidate models, PN is the number of predictions and PT is the computational time for behavior prediction.

TABLE III. COMPUTATIONAL COST TAKEN BY ETL-P AND MOD-ALL (MT: CANDIDATE MODELS TRAINING TIME; PN: BEHAVIOR PREDICTION NUMBERS; PT: BEHAVIOR PREDICTING TIME).

Per Exp.		eTL-P				Mod-All
#	Metrics	Navigator1	Navigator2	Navigator3	Avg	Avg
1	MT (ms)	46	53	325	141	419
2	PN	11688	14037	18787	14837	17558
3	PT (s)	2.55	3.43	51.10	19.02	41.90

When referring to MT, Mod-All tends to be more computationally expensive than eTL-P. According to the result in Table III, eTL-P reported an average MT of 141(ms), which is approximately one third that of Mod-All at 419(ms). That is expected since each navigator in Mod-All employs all candidate models while navigators in eTL-P work only with partial candidates via the multi-agent TL scheme. Further, when we consider a rather complex learning system with a larger agent size, multi-agent TL exhibits significantly better scalability. For example, when 10 agents are involved in the system, eTL-P reduces the training cost to a tenth of the required amount for Mod-All.

In addition, eTL-P has been observed to outperform Mod-All by reporting lower computational cost in terms of PT. According to the result attained by eTL-P, a significant cost reduction of 54.61% in average PT as compared to Mod-All is observed per experiment (see Table III, row 3). This is due to the reason that, navigators with multi-agent TL in eTL-P take advantage form their partners that are attributed with more reliable and effective predictive models. For example, with respect to PN, navigators with candidate model of high ρ^k tend to share the predicted information on predators to other navigators and hence obtains the most prediction number (18787, see Table III, row 2), much more than those obtained by navigators with low ρ^k and medium ρ^k (11688 and 14037, see Table III, row 2).

In summary, the proposed eTL-P with behavior prediction and multi-agent TL scheme has demonstrated its superiority in improving the success rate of navigators while also reducing the computational cost efficiency.

2) Performance of Top-K model selection scheme: Further, we proceed to validate the scalability of eTL-P in a complex scenario wherein navigators are confronted with greater number of predictive candidate choices. Since the search space of candidate models could be large, we endeavor to mitigate the model complexity of navigators' behavior prediction process based on the proposed Top-K model selection scheme described in Section III-C.

In this set of experiments, we consider 99 FALCON models as the predictive candidates which have differing vigilance values ρ^k at 0.01 intervals from $\rho^k = (0.01, 0.01, 0.01)$ to $\rho^k = (0.99, 0.99, 0.99)$. For comparison consideration, we firstly classify all 99 candidate models into categories of *1)* a low parameter set $\rho^k \in [0.01, 0.33]$, *2)* a medium parameter set $\rho^k \in [0.34, 0.66]$ and *3)* a high parameter set $\rho^k \in [0.67, 0.99]$, respectively. In this case, each navigator in eTL-P is attributed with a differing category of 33 predictive models. Further, to reduce the model complexity, we facilitate to select K models out of all 99 candidates using the proposed Top-K model selection method wherein the monotone submodular function considers a fusion of coverage and diversity principles. Particularly, a direct K-medoids clustering algorithm, where $K = \lfloor 0.1N \rfloor$ and N is the size of all candidates, is employed to enforce diversity principle as discussed in Section III-C. Hence, Top-K model selection method chooses 9 models with the largest representativeness from the full model space. Each navigator in eTL-P with Top-K models is merely assigned with 3 out of 99 candidate models.

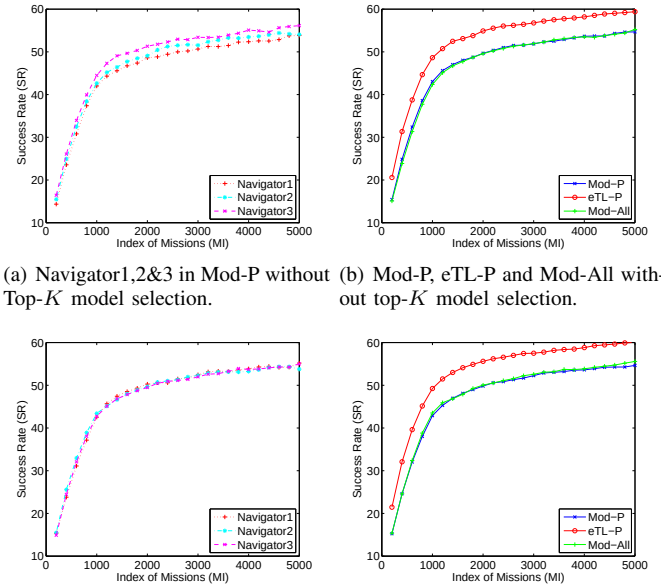
Table IV summarizes the complete results obtained pertaining to the success rates of Mod-P, eTL-P and Mod-All with all 99 candidate models or the selected Top-K models, at the end of 5,000 missions. Their corresponding learning performance is depicted in Fig. 7. It can be observed that overall, eTL-P with Top-K models outperformed most of its counterparts throughout the learning process. At the end of the 5,000 missions, the highest average SR of around 60% (see Table IV, column 8) has been attained, which is significantly higher than that of Mod-P and Mod-All.

Notably, navigators with Top-K models achieved competitive performance in terms of SR to navigators that consider all 99 candidate models. In particular, when attributed with a differing subset of Top-K candidate models, all navigators in Mod-P exhibit a consistently and high SR throughout the missions (see Fig. 7(a) and Fig. 7(c)). As discussed, this can be attributed to the proposed monotone submodular function in Top-K model selection wherein both coverage and diversity principles are taken into consideration.

Moreover, to validate and assess the efficacy of coverage and diversity principles, here we investigated differing monotone submodular functions, namely MS-C which only considers coverage principle ($\mathcal{F}(M^K) = \sigma(M^K)$) and MS-CD which further enforces diversity principle beyond the selections of MS-C ($\mathcal{F}(M^K) = \sum_{i=1}^{K} \sqrt{\sigma(P_i \cap M^K)}$), respectively. In order to evaluate MS-C and MS-CD quantitatively, we defined

TABLE IV. PERFORMANCE COMPARISON AMONG MOD-P, ETL-P AND MOD-ALL WITH TOP-K MODEL SELECTION SCHEME IN TERMS OF SR (SR: SUCCESS RATE).

	FALCON		SR (99 candidate models)			SR (Top-K models)		
#	Model Set	ρ^k	Mod-P	eTL-P	Mod-All	Mod-P	eTL-P	Mod-All
1	Low	$[0.01, 0.33]$	53.81	58.56		55.18	59.90	
2	Medium	$[0.34, 0.66]$	54.06	59.33	55.23	54.25	59.68	55.58
3	High	$[0.67, 0.99]$	56.13	60.28		55.02	60.55	



(a) Navigator1,2&3 in Mod-P without Top-K model selection.

(b) Mod-P, eTL-P and Mod-All without top-K model selection.

(c) Navigator1,2&3 in Mod-P with Top-K model selection.

(d) Mod-P, eTL-P and Mod-All with Top-K model selection.

Fig. 7. SRs of navigators with 99 predictive candidate models on completing the missions in PO-MNTs.

an efficiency ratio as follows:

$$\text{Ratio(Mod-P)} = \frac{\text{SR(Mod-P)} - \text{SR(Conv-MAS)}}{\text{MT} + \text{PT}} \tag{9}$$

where $[\text{SR(Mod-P)} - \text{SR(Conv-MAS)}]$ indicates the SR improvement of navigators exhibited by behavior prediction over Conv-MAS, and $[\text{MT} + \text{PT}]$ is the corresponding computing cost. Typically, a higher Ratio value is preferred since it denotes that the behavior prediction is more efficient by attaining higher SR improvement at a lower computational cost.

TABLE V. PERFORMANCE COMPARISON AMONG MOD-P WITH MS-C AND MS-CD (MT: CANDIDATE MODELS TRAINING TIME; PN: BEHAVIOR PREDICTION NUMBERS; PT: BEHAVIOR PREDICTING TIME).

	Mod-P	Metrics (per experiment)			
#	$\mathcal{F}(M^K)$	SR(%)	MT(ms)	PT(s)	Ratio(Mod-P)
1	MS-C	55.76	729.23	35.65	0.37
2	MS-CD	55.36	489.55	25.31	0.51

The complete results of MS-C and MS-CD are summarized in Table V. As we can see, although MS-C achieved a competitive SR of 55.76% to MS-CD of 55.36%, it incurred a significantly higher computational cost (i.e., PT and MT) during the learning process (see Table V, column 4 and 5). This is because MS-C is expected to select models that have a larger joint-coverage of all solutions. It prefers the candidate models with higher vigilance values which tend to contain an unnecessarily large number of redundant rules since knowledge generated becomes over-specific (see Section II-B).

On the other hand, MS-CD reduces the redundancy of the similar or overlapping information by enforcing the diversity principle in model selection. According to Table V, MS-CD reported a Ratio(Mod-P) of 0.51, which is significantly higher than 0.37 of MS-C (see Table V, column 6). Moreover, Fig. 8 denotes the learning performance of MS-C and MS-CD in terms of averaged success rate over computational cost ([MT+PT]). As observed, MS-CD tends to achieve higher SR against MS-C under the same computing cost. This result thus highlights the efficacy of the proposed monotone submodular function, which considers a fusion of coverage and diversity objectives, so as to provide navigators the option of selecting more representative Top-K candidates.

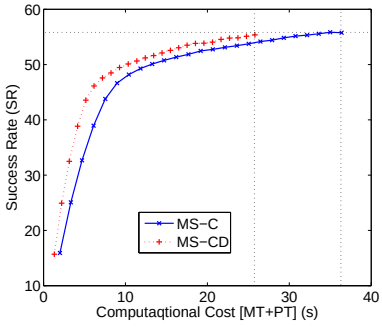


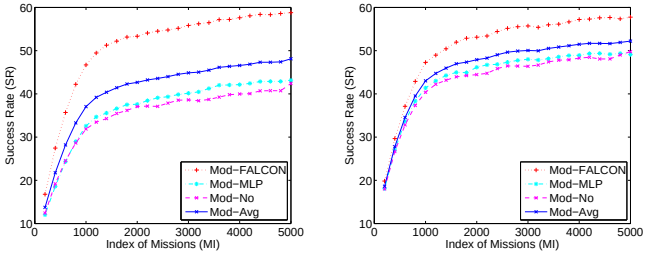
Fig. 8. SRs of MS-C and MS-CD over computational cost ([MT+PT]) on completing the missions in PO-MNTs.

E. Results of eTL-P with Predictive Candidates of Heterogeneous RL Structures

In previous subsections, we investigated the performance of eTL-P with homogeneous predictive candidates, i.e., we assume all candidate models taking the form of FALCON learning structures. In this section, we further showcase a common scenario involving predictive predator candidates of heterogeneous reinforcement learning structures. Specifically, both FALCON and a classical multi-layer perceptron (MLP) with gradient descent based back propagation [49]

are employed as the predictive candidates. For a fairness consideration, the configurations of FALCON and MLP are maintained to be consistent with the study in [18]. Similarly, both FALCON and MLP are trained using the data instances of predators recorded from historical activities of missions encountered.

Firstly, to investigate the performance of heterogeneous RL machines as candidate models in predicting the behavior of predators, we assign three navigators with FALCON (Mod-FALCON), MLP (Mod-MLP) and without candidate models (Mod-No), respectively. Also, their average results have been reported by Mod-Avg. Then, the learning performance of navigators in terms of SR is depicted in Fig. 9(a).



(a) Mod-FALCON, MLP, No&Avg. (b) Mod-FALCON, MLP, No&Avg with eTL-P.

Fig. 9. SRs of Mod-FALCON, Mod-MLP, Mod-No and Mod-Avg on completing the missions in PO-MNTs.

As can be observed, Mod-FALCON shows its superiority in attaining much higher SR than Mod-MLP and Mod-No. This is not surprising as FALCON is the exact learning model of predators in PO-MNT. After training with the collected data instances, it could predict the behaviors of predators in a comparatively accurate manner.

Subsequently, to access the effectiveness of our proposed approach, SRs of Mod-FALCON, Mod-MLP and Mod-No with eTL-P are depicted in Fig. 9(b). We can see the proposed eTL significantly improved the overall learning performance of three navigators in terms of SR. In particular, Mod-MLP and Mod-No achieved approximately 6.0% and 7.3% improvements in SR, respectively. The result hence demonstrates the efficacy of proposed eTL-P in improving the learning performance of navigators using heterogeneous RL techniques as the candidate models in PO-MNT.

V. Conclusion

This paper has presented an enhanced evolutionary Transfer Learning framework, eTL-P, for addressing the specific challenges that arise in complex multi-agent systems where agents have competing objectives. Particularly, by providing a behavior prediction approach, eTL-P endows agents with abilities to predict the behaviors of opponent agents effectively by building the candidate models. In order to reduce the complexity or computational cost of behavior prediction, eTL-P proposes a monotone submodular function considering both coverage and diversity functional objectives, and further introduces a multi-agent TL scheme to select a representative and much smaller subset of candidate models from the full model space. The performance efficacy of eTL-P is investigated via comprehensive empirical studies in a PO-MNT. Accordingly, eTL-P can significantly reduce the complexity of candidate models while improve the learning capability of multiple agents in an effective manner.

Generally, model-based behavior prediction, Top-K model selection and multi-agent TL comprise the core learning components of eTL-P. In the immediate future, we would like to explore the generality and adaptivity of the proposed eTL-P in solving the increasing complexity and diversity of problem solving, by focusing on the novel on-line model predicting approaches, model-based or model-free behavior approximating methods, representative model selection functions as well as other knowledge transfer approaches.

References

[1] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *The Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.

[2] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 951–958.

[3] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 1225–1237.

[4] Y. Hu, Y. Gao, and B. An, "Accelerating multiagent reinforcement learning by equilibrium transfer," *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1289–1302, 2015.

[5] G. Acampora, J. M. Cadenas, V. Loia, and E. M. Ballester, "A multi-agent memetic system for human-based knowledge selection," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 5, pp. 946–960, 2011.

[6] W. Wang, A.-H. Tan, and L.-N. Teow, "Semantic memory modeling and memory interaction in learning agents," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 11, pp. 2882–2895, 2017.

[7] E. Oliveira and L. Nunes, "Learning by exchanging advice," *In R.Khosla, N. Ichalkaranje, and L. Jain, editors, Design of Intelligent Multi-Agent Systems, chapter 9. Spring, New York, NY, USA*, 2004.

[8] A. Taylor, I. Dusparic, E. Galván-López, S. Clarke, and V. Cahill, "Transfer learning in multi-agent systems through parallel transfer," in *Theoretically Grounded Transfer Learning at the 30th International Conference on Machine Learning (ICML). Omnipress*, 2013.

[9] Y. Hou, Y.-S. Ong, L. Feng, and J. M. Zurada, "An evolutionary transfer reinforcement learning framework for multi-agent system," *IEEE Transactions on Evolutionary Computation*, 2017.

[10] D. Yang, X. Chen, and J. Jiang, "Multi-agent cooperation based on behavior prediction and reinforcement learning," in *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, vol. 6. IEEE, 2004, pp. 4869–4872.

[11] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *arXiv preprint arXiv:1709.08071*, 2017.

[12] S. C. Bakkes, P. H. Spronck, and G. van Lankveld, "Player behavioural modelling for video games," *Entertainment Computing*, vol. 3, no. 3, pp. 71–79, 2012.

[13] I. Dagli and D. Reichardt, "Motivation-based approach to behavior prediction," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 227–233.

[14] Y. Hou, Y. Zeng, and Y.-S. Ong, "A memetic multi-agent demonstration learning approach with behavior prediction," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 539–547.

[15] F. Hutter, "Automated configuration of algorithms for solving hard computational problems," Ph.D. dissertation, University of British Columbia, 2009.

[16] Y.-S. Ong, M. H. Lim, and X. Chen, "Research frontier-memetic computation-past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, p. 24, 2010.

[17] R. Dawkins, "The selfish gene," *Oxford: Oxford University Press*, 1976.

[18] A.-H. Tan, N. Lu, and D. Xiao, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *Neural Networks, IEEE Transactions on*, vol. 19, no. 2, pp. 230–244, 2008.

[19] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[20] H. Kebriaei, A. Rahimi-Kian, and M. N. Ahmadabadi, "Model-based and learning-based decision making in incomplete information cournot games: A state estimation approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 4, pp. 713–718, 2015.

[21] A. Konar, I. Goswami, S. J. Singh, L. C. Jain, and A. K. Nagar, "A deterministic improved q-learning for path planning of a mobile robot." *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1141–1153, 2013.

[22] D. Liu, X. Yang, D. Wang, and Q. Wei, "Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints," *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1372–1385, 2015.

[23] A. Lazaric, M. Restelli, and A. Bonarini, "Reinforcement learning in continuous action spaces through sequential monte carlo methods," in *Advances in neural information processing systems*, 2007, pp. 833–840.

[24] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine learning*, vol. 22, no. 1-3, pp. 123–158, 1996.

[25] M. P. Deisenroth, G. Neumann, J. Peters *et al.*, "A survey on policy search for robotics." *Foundations and Trends in Robotics*, vol. 2, no. 1-2, pp. 1–142, 2013.

[26] M. E. Taylor, N. K. Jong, and P. Stone, "Transferring instances for model-based reinforcement learning," in *Machine learning and knowledge discovery in databases*. Springer, 2008, pp. 488–505.

[27] M. E. Taylor and P. Stone, "Representation transfer for reinforcement learning," in *AAAI 2007 Fall Symposium on Computational Approaches to Representation Change during Learning and Development*, 2007.

[28] Y. Hu, Y. Gao, and B. An, "Multiagent reinforcement learning with unshared value functions," *IEEE transactions on cybernetics*, vol. 45, no. 4, pp. 647–662, 2015.

[29] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 617–624.

[30] X. Chen, Y. Zeng, Y.-S. Ong, C. S. Ho, and Y. Xiang, "A study on like-attracts-like versus elitist selection criterion for human-like social behavior of memetic mulitagent systems," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1635–1642.

[31] T. Nierhoff, K. Leibrandt, T. Lorenz, and S. Hirche, "Robotic billiards: Understanding humans in order to counter them," *IEEE transactions on cybernetics*, vol. 46, no. 8, pp. 1889–1899, 2016.

[32] G. Kuhlmann, W. B. Knox, and P. Stone, "Know thine enemy: A champion robocup coach agent," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1463.

[33] P. Stone, "Learning and multiagent reasoning for autonomous agents." in *IJCAI*, 2007, pp. 12–30.

[34] R. Mealing and J. L. Shapiro, "Opponent modeling by expectation–maximization and sequence prediction in simplified poker," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 1, pp. 11–24, 2017.

[35] S. V. Albrecht and S. Ramamoorthy, "A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1155–1156.

[36] S. Barrett, P. Stone, and S. Kraus, "Empirical evaluation of ad hoc teamwork in the pursuit domain," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 567–574.

[37] S. Barrett, P. Stone, S. Kraus, and A. Rosenfeld, "Teamwork with limited knowledge of teammates." in *AAAI*, 2013.

[38] S. V. Albrecht, J. W. Crandall, and S. Ramamoorthy, "An empirical study on the practical impact of prior beliefs over policy types." in *AAAI*, 2015, pp. 1988–1994.

[39] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Machine learning*, vol. 49, no. 2-3, pp. 161–178, 2002.

[40] M. E. Taylor, N. K. Jong, and P. Stone, "Transferring instances for model-based reinforcement learning," in *Machine learning and knowledge discovery in databases*. Springer, 2008, pp. 488–505.

[41] R. Conroy, Y. Zeng, and J. Tang, "Approximating value equivalence in interactive dynamic influence diagrams using behavioral coverage," 2016.

[42] Y. Zeng and P. Doshi, "Exploiting model equivalences for solving interactive dynamic influence diagrams." *J. Artif. Intell. Res.(JAIR)*, vol. 43, pp. 211–255, 2012.

[43] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[44] H. Lin, J. Bilmes, and S. Xie, "Graph-based submodular selection for extractive summarization," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 381–386.

[45] G. Cornuejols, M. Fisher, and G. L. Nemhauser, "On the uncapacitated location problem," *Annals of Discrete Mathematics*, vol. 1, pp. 163–177, 1977.

[46] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*. Springer, 1983, pp. 235–257.

[47] Y. Zeng, X. Chen, Y.-S. Ong, J. Tang, and Y. Xiang, "Structured memetic automation for online human-like social behavior learning," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 102–115, 2017.

[48] A.-H. Tan and D. Xiao, "Self-organizing cognitive agents and reinforcement learning in multi-agent environment," in *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*. IEEE, 2005, pp. 351–357.

[49] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, U.K, May 1989.